Enhancing Concept Detection by Pruning Data with MCA-based Transaction Weights

Lin Lin, Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL 33124, USA
Email: l.lin2@umiami.edu, shyu@miami.edu

Shu-Ching Chen
School of Computing and Information Sciences
Florida International University
Miami, FL 33199, USA
Email: chens@cs.fiu.edu

***Abstract*—With the rapid increase in the amount of multimedia data, the researches on semantic information retrieval are facing a very challenging problem - the number of positive data instances with the target concept/object/event compared with the number of negative data instances without the target concept/object/event is much smaller, which is also called the data imbalance issue. Therefore, one of the popular topics in multimedia information processing and retrieval is data pruning, a technique that can automatically identify and prune the data instances from the training data set so that the pruned data set is able to enhance the performance of model learning, classification, and concept detection. In this paper, a novel data pruning framework which gives each transaction a weight based on multiple correspondence analysis (MCA) is proposed. These transaction weights are used as the measure for pruning the training data set. Meanwhile, the testing data set could be weighted and pruned as well so that the computational cost is reduced not only when building the model but also when applying the classifiers. Experimenting with 18 high-level concepts and the benchmark (both balanced and imbalanced) data sets from TRECVID, our proposed framework achieves promising results to enhance the concept detection performance of three well-known classifiers commonly used for concept detection.**

***Keywords*-data pruning; transaction weight; concept detection; multiple correspondence analysis.**

I. Introduction

With the advanced techniques of digital record devices, the fast development of the Internet platform, and the low cost of storage devices, it becomes much easier to distribute and collect the multimedia data nowadays. The rapid increase in the amount of the multimedia data, the inefficient traditional text-based information retrieval approaches, and the high demands for the multimedia analysis and management applications have motivated the researchers to devote into the area of content-based multimedia retrieval [1] [2][3][4].

The training set is very important for multimedia retrieval and classification, since the use of a good sampling data set would potentially influence the results significantly [5]. However, the data size is usually very large and the number of positive data instances of the target concept, object, or event is quite insufficient in the multimedia database. When there are too many "noisy" data instances in the training data set, they can mislead the algorithm, create confusing boundaries, and increase the complexity. Even for some algorithms that have the abilities to ignore the noisy data instances and optimize the mechanisms or training models, e.g., Support Vector Machine (SVM) and Neural Network, there are data instances adversary to learning and thus the performance is still poor. Therefore, the data imbalance issue is considered as one of biggest challenges [6][7]. To address the data imbalance issue, the *data pruning* process/technique can be utilized in the manner that given the training data set and the learning model, it can reduce the data set and select the representative data instances as the new training data set so that the performance of the model and the classification result would be improved.

For traditional document retrieval, information filtering is either content-based or collaborative one. The content-based method is usually based on the term frequency of text documents and the collaborative filtering method is based on the particular user's selection. The filtering algorithms integrated the content-based filtering and collaborative filtering are studied for the audio, image and video media in the later years [8]. For pruning multimedia data, the most simplest method to select the sample data set would be simple random sampling (SRS). However, this leads to unsatisfactory results due to the fact that the random samples (data instances) may not adequately represent the entire data set. Researchers have been investigating several categories of approaches to select the samples and prune the "meaningless data instances" for the learning model [9][10].

One category of techniques is the probability-based. The methods in this category prune the samples that could be identified by estimating the posterior probability of the possible label of each sample. Angelova et al. [11] first introduced data pruning to computer vision community and used the Naive Bayes approach to estimate the posterior probability of the label given a sample. The data instance whose estimated label is different from the original label is considered as the data instance to be pruned. The authors in [12] proposed an algorithm to remove the confusing data instances by studying the behavior of AdaBoost trained on the resulting data set. They used Platt scaling to convert the estimated labels directly into the averaged posterior probabilities of the label of each data instance. Herman et al. [13] presented that data pruning algorithm based on the recursive Bayes approach and AdaBoost could increase the confidence of the predictions in every iteration and minimize the number of predictions that have low confidence.

Another category is the distance-based data pruning techniques that the detection of the data instances to be pruned is based on the distance measure. In [14], a simple distance measure that compares the histograms of the sample data set with those of the whole training data set was proposed, in order to estimate the representativeness of the sample data set. Experimented on image and audio data, the results showed an elegant manner of noise reduction. In [15], the geodesic distances between pairs of data instances were used to subsample. The estimated geodesic distances calculated by Isomap were sorted in an increasing order. If any geodesic distance between the current data instance and the others was smaller than a given threshold, then that data instance would be deleted. The value of the threshold was determined based on the number of the pruned data instances. In [16], only the objects' mutual distances were utilized and it did not need to place any constraint on the data and the distance function.

The third category of the data pruning techniques is density-based. Unlike the distance-based methods that have limited abilities of handling the data sets with varying densities, the density-based approaches on the other hand have the ability to do so. The one relied on the Local Outlier Factor (LOF) of each object [17] was the most influential approach. The LOF of each object was based on the density of an object's neighborhood which specified the minimum number of objects in the nearest neighborhood. The objects to be pruned were with a high LOF.

The fourth category is the clustering-based data pruning techniques, where the detection of the data instances to be pruned is based on the clustering process. Xiong et al. [18] proposed the hyperclique cleaner (HCleaner) based on the hypothesis that once the data is clustered, the noisy objects are the farthest from their corresponding cluster centroids. The HCleaner used the hyperclique patterns as a filter to eliminate data objects that were not tightly connected to other data objects in the data set. Sarawagi et al. [19] discussed a Top-K method of collapsing and pruning data instances. The Top-K queries returned the set of K largest groups in the data, and the data instances that were guaranteed not to belong to any of the Top-K groups were pruned.

In this paper, we propose a novel correlation-based data pruning approach. In our previous study [20][21], we have utilized the *Multiple Correspondence Analysis (MCA)* methodology to capture the correlation between the features and the classes, and generate the association rules for classification. MCA is an extension of standard correspondence analysis to more than two variables [22]. While considering a multimedia database, the columns represent the features and classes and the rows represent the data instances. This correlation information gives us important knowledge such as (1) each transaction has an un-equal weight, (2) different features demonstrate different weight contributions, and (3) even for the same feature, it has different weight contributions to different classes. Here, a transaction is defined as a data instance consisting of the feature-value pairs discretized from the low-level features. For the training data, a transaction is a data instance without the class label; while for the testing data, since the class label is unknown, a transaction is a data instance. Therefore, MCA is applied to weight each transaction. Each transaction in the multimedia database is associated with an initial weight of 1. The transaction weight is updated with the sum of the correlation information of each feature obtained from MCA. The algorithm is able to automatically and adaptively choose (1) the threshold for the weight so that only the features best represent the data would be weighted, and (2) the threshold for the total transaction weight to determine whether the data instance should be pruned or not.

To evaluate our proposed framework, we used 18 high-level concepts and (both balanced and imbalanced) data sets from TRECVID 2007 and 2008 [23]. The performance evaluations of three well-known classifiers, namely the *Decision Tree* (DT), *Support Vector Machine* (SVM), and *Neural Network* (NN), trained by the training data set before and after pruning, are presented to demonstrate the efficiency and effectiveness of our proposed MCA-based data pruning framework. The reasons these three classifiers were used are that they are the most popular classifiers in the TRECVID community. Overall, our proposed framework is able to prune around 50% of the negative (non-concept) data instances from the balanced data sets and about 80% of the negative (non-concept) data instances from the imbalanced data sets. The performance (recall, F1-score, average accuracy) of the classifiers increases more than 10% using the pruned data sets for all 18 concepts.

This paper is organized as follows. In Section II, the proposed framework is presented and detailed discussion on each component is provided. The discussions on the experiments as well as the result analysis are given in Section III. Section IV concludes this paper.

II. The Proposed MCA-based Data Pruning Framework

A novel data pruning framework that utilizes the MCA-based transaction weights is proposed. The system architecture of our proposed MCA-based data pruning framework is shown in Figure 1. As can be seen from this figure, our proposed framework is composed of three components, namely *Data Preparation Component*, *Data Pruning Component*, and *Model Training and Classification Component*.

In the data preparation component, a set of 28 low-level audio-visual numeric features is extracted from the video data and the normalization process is applied to scale all numeric values (skipping the class label feature) in the data set to lie between zero and one. The normalization technique adopted here is to subtract the minimum value and divide by the range between the maximum and the minimum values for each features. The normalized data is then split into a training data set (two thirds of the whole data) and a testing data set (one third of the whole data). In order to apply MCA properly, each feature in the training data set is discretized into several partitions (i.e., feature-value pairs), and the same partition ranges are used to discretize the testing data set. After the discretization process, both the discretized training and testing data sets are passed to the data pruning component.

In the data pruning component, we first apply MCA to the discretized training data set to compute the MCA-based transaction weights (to be detailed in Section II-A). Using the transaction weights, the thresholds for data pruning can be determined. In order to obtain the best thresholds, an iterative process is carried out (the dashed lines in Figure 1 and to be discussed in Section II-B). The best thresholds obtained will be utilized to prune the discretized testing data set.

Finally, the model training and classification component takes the pruned training data set to train/build the classifier, and the trained classifier takes the pruned testing data set for concept detection (classification). The classification results (i.e., high-level concept detection) are then evaluated using the precision, recall, and F1-score metrics.

A. MCA-based Transaction Weights

After discretization, the numeric data instances in the training data set are converted to nominal data instances. For example, the low-level feature values in the training data set shown in Table I can be discretized to the feature-value pairs shown in Table II. Each numeric feature has various nominal feature-value pairs A_i^j (where j=1 to the total number of partitions for the ith feature and i=1 to 28). For instance, $feature_{17}$ (i.e., i=17) is the feature of the pixel changes, and it is converted to 3 partitions by applying discretization (i.e., j=1, 2, or 3). Feature-value pairs (A_{17}=A_{17}^1), (A_{17}=A_{17}^2), and (A_{17}=A_{17}^3) represent the partitions of the feature value ranges [0, 0.32865], (0.32865, 0.5044], and (0.5044, 1], respectively.

The multiple correspondence analysis could be used to capture the correlation among more than two variables in the table [22]. In our study, each training data instance is one row in the table and is represented by the values of all the

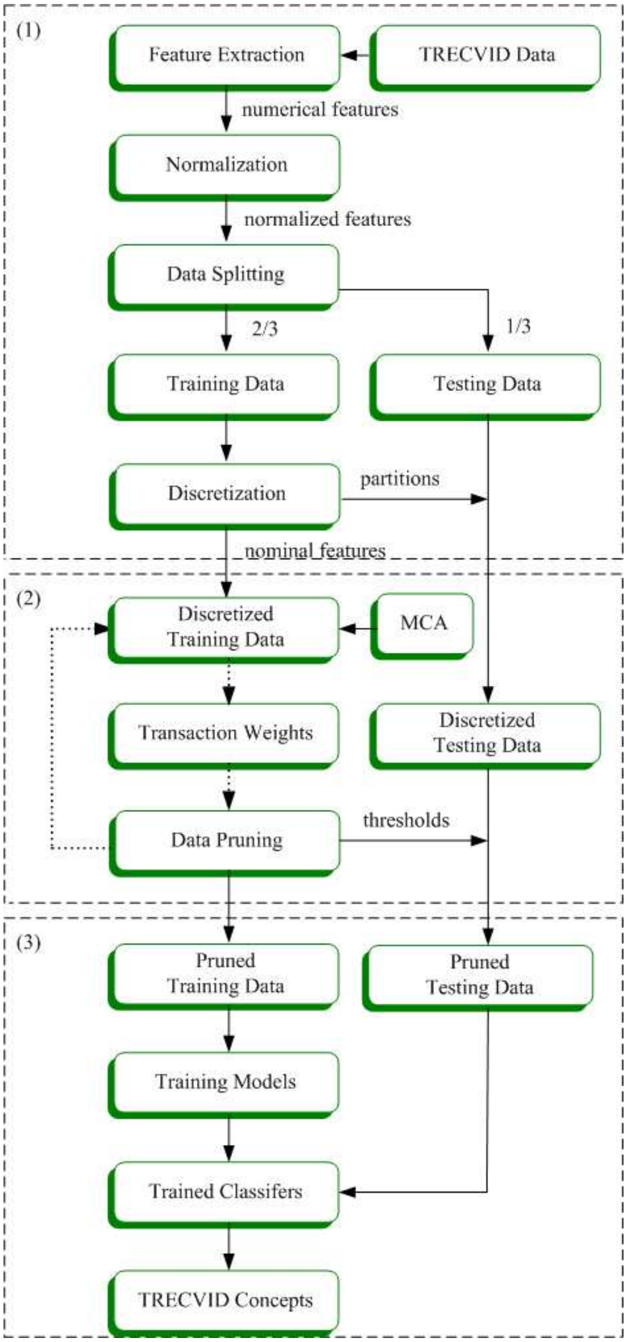

Figure 1. System architecture of the proposed MCA-based data pruning framework

Table I
EXAMPLES FOR NUMERIC DATA INSTANCES IN THE MULTIMEDIA DATABASE

$feature_1$ (A_1)	$feature_2$ (A_2)	... ...	$feature_{28}$ (A_{28})	$class$
0.23	0.78	...	0.05	C_p
0.10	0.70	...	0.18	C_n
...	...	...	...	...
0.17	0.67	...	0.78	C_n
...	...	...	...	...

Table II
EXAMPLES FOR NOMINAL DATA INSTANCES IN THE MULTIMEDIA DATABASE

$feature_1$ (A_1)	$feature_2$ (A_2)	... ...	$feature_{28}$ (A_{28})	$class$
A_1^2	A_2^3	...	A_{28}^1	C_p
A_1^1	A_2^3	...	A_{28}^2	C_n
...	...	...	...	...
A_1^1	A_2^3	...	A_{28}^4	C_n
...	...	...	...	...

feature-value pairs belonging to the features together with the class label feature in the columns of the table. Therefore, MCA is applied to project the feature value space into the principle component space and to calculate the correlation information between each feature-value pair and each class for the training data instances. From the calculation of the inner product of a feature-value pair (A_i^j) and class (C_p or C_n), i.e., the cosine of the angle, the angle ($angle_i^j \in [0, 180]$) could be captured as a measurement to represent the correlation, where i is from 1 to 28, j varies for each feature from 1 to the number of partitions for each feature, C_p is the target concept class, and C_n is the non-concept class.

If $angle_i^j$ between feature-value pair A_i^j and the class label C_p is smaller than 90 degrees, it shows that the feature-value pair has a higher correlation relationship with the positive class. If $angle_i^j$ between the feature-value pair A_i^j and the class label C_n is smaller than 90 degrees, it demonstrates that the feature-value pair indicates a higher correlation relationship with the negative class. If $angle_i^j$ is equal to 90 degrees, it means that the feature-value pair has equal correlation relationships with both positive and negative classes.

Table III
EXAMPLE TRANSACTIONS IN THE TRAINING DATA SET

$feature_1$ (A_1)	$feature_2$ (A_2)	... ...	$feature_{28}$ (A_{28})
A_1^2	A_2^3	...	A_{28}^1
A_1^1	A_2^3	...	A_{28}^2
...	...	...	...
A_1^1	A_2^3	...	A_{28}^4
...	...	...	...

In our proposed framework, a transaction in the training data set is defined as a data instance without its corresponding class label. Table III gives some example transactions in the training data set. The transaction weight for each data instance, $TWeight_k$ (shown in Equation (1)), is calculated by the sum of its feature-value pair weights $weight_i^j$ (shown in Equation (2)), where k=1 to the total number of data instances, i=1 to 28 features, and j can be any value between 1 and the number of partitions for the i^{th} feature. In addition, when $angle_i^j = 90$, we set $weight_i^j = 0$. When $angle_i^j$ between A_i^j and C_p is less than 90, then positive sign is set. Otherwise, when $angle_i^j$ between A_i^j and C_n is less than 90, then the negative sign is set. Then $TWeight_k$ value can be calculated. Table IV shows some examples.

$$TWeight_k = \sum_{i=1}^{28} weight_i^j; \tag{1}$$

$$weight_i^j = \pm(180 - angle_i^j)/90. \tag{2}$$

Table IV
EXAMPLES FOR FEATURE-VALUE PAIR WEIGHTS AND TRANSACTION WEIGHTS

$feature_1$ (A_1)	$feature_2$ (A_2)	... ...	$feature_{28}$ (A_{28})	$TWeight_k$
1.02	1.32	...	1.95	3.256
-1.05	1.32	...	-1.05	-1.360
...	...	...	...	...
-1.05	1.32	...	-1.92	-2.123
...	...	...	...	...

Note that in the traditional multimedia databases, the data instances are usually considered to be equal, and the feature-value pairs in each data instance are also considered equally important with respect to the class labels. On the other hand, when applying the strategy of MCA-based transaction weights to the training data, the transaction weights of the data instances can have different values, and the feature-value pair weights can fall into the range of $[-2, 2]$ (i.e., $weight_i^j \in [-2, 2]$). In other words, by utilizing MCA, the feature-value pairs can be weighted differently based on their correlation to the various class labels. Note that due to our definition of the transaction weights, when the absolute values of the corresponding weight values are large, the data instances are considered as typical/pure positive or negative data instances. Therefore, the transaction weights can be used as a measure to identify the noisy data instances.

B. *Threshold Determination*

After getting the weight for each feature-value pair, the thresholds for the weights that can capture the best representative feature-value pairs for each class should be determined. In our proposed framework, the non-zero weights are sorted in the ascending order without duplicate values for each class label, and an iterative process is carried out in order to find the best threshold value for the training data set. In each iteration, a different threshold value is applied to the training data set and the one with the highest F1-score is selected as the threshold value.

The iteration for positive weight threshold starts from the second smallest value to the number of the sorted non-zero positive weights without duplicate, which ensures that at least one positive weight would be kept. The iteration for negative weight threshold starts from 1 to the number of sorted non-zero negative weights without duplicate. Every feature-value pair weight is checked so that if $weight_i^j$ is higher than the positive weight threshold or $weight_i^j$ is lower than the negative weight threshold, then it is kept. Otherwise, the weight is set to 0. Using these updated feature-value pair weights, the transaction weights of the training set can be obtained.

Let $mean_{neg}$ be the mean of the transaction weights for those transactions belonging to the non-concept class, and std_{neg} be the standard deviation of the transaction weights for those transactions belonging to the non-concept class. $index$ is from 0 to 49, which gives the biggest value of this threshold, $mean_{neg} + std_{neg}$, and the smallest value of this threshold, $mean_{neg} + std_{neg} \times 0.08$. From our empirical study, $std_{neg} \times 0.08$ is small enough to be around 1. The algorithm can also automatically and adaptively choose another threshold (i.e., $threshold3$ defined in Equation (3)) for the total transaction weight to determine whether the data instance should be pruned or not. In order to determine $threshold3$, another iteration with applying different threshold values (i.e., varying the $index$ values from 0 to 49) to the training data set is processed, and the $threshold3$ value which yields the highest F1-score will be used.

$$threshod3 = mean_{neg} + std_{neg}/(1 + index/4). \qquad (3)$$

Now, our proposed data pruning strategy works as follows. If a transaction weight is larger than or equal to this $threshold3$ value, the transaction is considered to be estimated as a positive data instance. On the other hand, if a transaction weight is smaller than this $threshold3$ value, the transaction is considered to be estimated as a negative data instance. Then the training data set is pruned so that only those true/pure positive and true/pure negative data instances that pass this $threshold3$ value will be kept in the pruned training data set.

Each testing transaction is checked whether the corresponding feature-value pairs exist and is weighted by the updated feature-value pair weights. The transaction weight is calculated and the same data pruning strategy is applied to these transactions. Similarly, the testing data set is pruned in the way that any data instance whose transaction weight is smaller than the $threshold3$ value is pruned as a negative instance. The remaining data instances are considered as the pruned testing data set.

The classification model (or classifier) will be trained using the pruned training data set, and the pruned testing data set will be classified using the trained classifier. Since both the training and testing data sets have been pruned, the computational costs of both learning and classification have been reduced. The final classification results consist of two parts. The first part is the misclassified positive data instances during the data pruning process for the testing data set and the correctly classified negative data instances. The second part includes the results generated by classifying the pruned testing data set using the trained classifier.

III. Experiments and Results

Our proposed framework is validated using the available data taken from the videos used in the TRECVID high-level feature extraction task in 2007 and 2008 [23]. The nine high-level concepts (or concept classes) which were used in our previous study [21] are hand, urban, crowd, person, two-people, outdoor, building, vegetation, and road. For these nine concepts, the data sets are considered as balanced since they were generated with the 50% ratio, including all the positive data instances and the randomly selected negative data instances that are twice of the size of the positive ones. The other nine high-level concepts which are considered as the imbalanced data sets include sports, cityscape, snow, nighttime, boat-ship, singing, police, military, and car. For these nine concepts, all the positive and negative data instances were used, and the ratios of positive vs negative data instances are all very small. The descriptions of these high-level concepts are available in [23].

To show the efficiency and effectiveness of our proposed framework, the performances of three well-known classifiers, namely the Decision Tree (C4.5 algorithm, denoted as DT), Support Vector Machine (Sequential Minimal Optimization (SMO) algorithm, denoted as SVM), and Neural Network (Multilayer Perception algorithm, denoted as NN) available in WEKA [24], trained by the original and pruned training data sets are compared. To evaluate the framework, the precision (pre), recall (rec), and F1-score (F1) performance metrics are adopted under the 3-fold cross-validation approach which ensures that each data instance in the data set would be tested in the experiment. In addition, the average accuracy values of the positive and negative concepts for the imbalanced data sets before and after pruning are also presented.

The results for the imbalanced data sets are shown in Table V. Columns 2 to 4 are the results before data pruning, and columns 5 to 7 are the results after data pruning. As can be seen from Table V, using the original imbalanced training data set, the models could not even be built since the precision, recall, and F1-score values are almost 0, but our proposed data pruning framework enhances the performance of DT, SVM, and NN classifiers with promising results. The better recall values represent that more positive data instances belonging to the concept class are classified correctly. It can also be observed that both the recall values and F1-scores of the classifiers trained by the pruned training data set are higher than those trained by the original training data set. The better F1-scores mean that the performances of the classifiers are improved without misclassifying too many negative data instances.

To further demonstrate the efficiency and effectiveness of our proposed framework, Table VI first shows the number of negative data instances (3rd column), the ratio of the number

Table V
PERFORMANCE EVALUATION FOR NINE IMBALANCED DATA SETS, WHERE COLUMNS 2 TO 4 ARE THE RESULTS BEFORE DATA PRUNING AND COLUMNS 5 TO 7 ARE THE RESULTS AFTER DATA PRUNING.

sports	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.00	0.17	0.33	0.11	0.10	0.07
Rec	0.00	0.01	0.00	0.18	0.24	0.49
F1	0.00	0.02	0.01	0.14	0.14	0.12
cityscape	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.00	0.00	0.00	0.06	0.05	0.04
Rec	0.00	0.00	0.00	0.26	0.11	0.27
F1	0.00	0.00	0.00	0.07	0.07	0.07
snow	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.00	0.00	0.00	0.16	0.10	0.07
Rec	0.00	0.00	0.00	0.47	0.60	0.66
F1	0.00	0.00	0.00	0.23	0.17	0.12
nighttime	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.00	0.35	0.45	0.18	0.18	0.09
Rec	0.00	0.13	0.04	0.45	0.50	0.64
F1	0.00	0.19	0.08	0.25	0.25	0.16
boat/ship	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.00	0.23	0.46	0.12	0.09	0.07
Rec	0.00	0.04	0.04	0.25	0.31	0.47
F1	0.00	0.06	0.07	0.16	0.14	0.33
singing	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.00	0.00	0.00	0.10	0.08	0.04
Rec	0.00	0.00	0.00	0.29	0.29	0.47
F1	0.00	0.00	0.00	0.15	0.13	0.08
police	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.00	0.00	0.00	0.15	0.12	0.06
Rec	0.00	0.00	0.00	0.19	0.24	0.47
F1	0.00	0.00	0.00	0.06	0.16	0.10
military	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.00	0.00	0.00	0.03	0.03	0.04
Rec	0.00	0.00	0.00	0.46	0.42	0.38
F1	0.00	0.00	0.00	0.06	0.06	0.07
car	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.00	0.00	0.00	0.08	0.09	0.06
Rec	0.00	0.00	0.00	0.08	0.16	0.29
F1	0.00	0.00	0.00	0.07	0.11	0.11

of positive data instances to the number of negative data instances (4th column), and the pruning rate of the negative data instances (5th column) for the imbalanced data sets. From this table, it can be clearly seen that the data sets are very imbalanced since the ratios are between 0.8% and 2.8% for these nine concepts. Furthermore, our MCA-based data pruning framework is able to effectively prune more than 80% of the negative data instances for each concept, which in turn reduces the complexity significantly to enhance the classification efficiency. For example, for the *snow* concept, the ratio between the positive data instances to the negative data instance is as low as 0.8% and our proposed framework is able to prune 84% of the negative data instances. As can be seen from Table V, such pruning improves the F1-score values of the SVM, DT, and NN classifiers from 0, 0, and 0 to 0.23, 0.17, and 0.12, respectively.

Next, the average accuracy results of the positive and negative classes using the original and pruned testing data sets for the nine concepts are shown in Figure 2. Here, SVM bp and SVM ap mean the results before pruning and after pruning for the SVM classifier. Similar notations apply to the DT and NN classifiers. It can be easily seen from the comparison results that the proposed framework enhances the average accuracy performance for all nine concepts under all three classifiers. For example, for the *snow* concept, the average accuracy for NN bp (NN classifier before pruning) is 0.50 (0.00 for the positive class and 1.00 for the negative class); while the average accuracy for NN ap (NN classifier after pruning) reaches 0.79 (0.66 for the positive class and 0.92 for the negative class).

Table VI
IMBALANCED DATA SETS – INFORMATION OF THE NINE INVESTIGATED CONCEPTS

	Concept	# of negative data instances	P / N ratio	Pruned negative data instances
1	sports	14586	0.015	83.0%
2	cityscape	14355	0.011	83.3%
3	snow	14691	0.008	84.0%
4	nighttime	14301	0.014	84.6%
5	boat/ship	14223	0.020	83.4%
6	singing	14379	0.009	83.4%
7	police	14655	0.011	85.0%
8	military	14619	0.013	82.2%
9	car	14409	0.028	83.0%

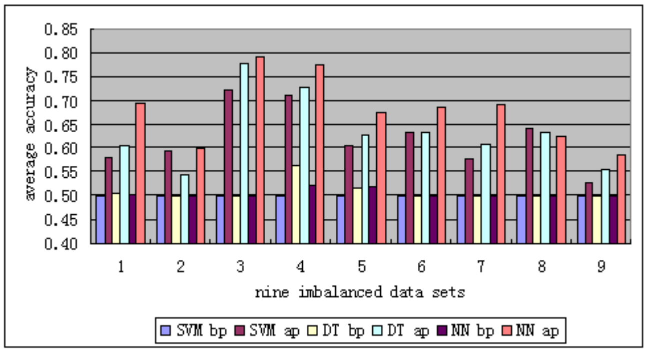

Figure 2. The average accuracy of positive and negative classes before and after pruning for imbalances data sets

In addition to the imbalanced data sets, our proposed framework is also tested on those nine concepts used in our previous study [21], which are considered as balanced data sets. The results are shown in Table VII. Similarly, in Table VII, columns 2 to 4 are the results before data pruning, and columns 5 to 7 are the results after data pruning. DT, SVM, and NN represent the Decision Tree classifier, Support Vector Machine classifier, and Neural Network classifier, while Pre, Rec, and F1 represent the precision, recall, and F1-score values.

It can be observed from Table VII that using the original balanced training data sets, most of the classifiers could build the models and give reasonable precision, recall, and F1-score values. Due to the trade-off between precision and recall, a better F1-score value is usually considered as better performance to demonstrate the accuracy of the proposed framework. On the other hand, the classifiers trained by using the pruned training data sets, are able to reach higher recall and F1-score values than those classifiers trained by the original data sets. This conclusion not only shows that our proposed pruning strategy assists the classifiers in identifying more positive data instances, but also shows that our pruning strategy does not prune too many negative data instances. Otherwise, if the negative data instances are pruned too much, the remaining negative data instances will not be able to build the model and thus most the data instances will be labeled as positive. Then even the recall value will be 1, the framework is not considered as a good one since the precision and F1-score will be very low in that case. Furthermore, this table demonstrates the robustness of our proposed framework which works for both imbalanced and balanced data sets.

Table VII
PERFORMANCE EVALUATION FOR NINE BALANCED DATA SETS, WHERE COLUMNS 2 TO 4 ARE THE RESULTS BEFORE DATA PRUNING AND COLUMNS 5 TO 7 ARE THE RESULTS AFTER DATA PRUNING

2 people	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.00	0.51	0.44	0.39	0.39	0.38
Rec	0.00	0.19	0.25	0.66	0.65	0.66
F1	0.00	0.27	0.32	0.49	0.48	0.48

outdoor	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.59	0.59	0.53	0.46	0.47	0.46
Rec	0.37	0.39	0.48	0.74	0.71	0.74
F1	0.46	0.47	0.50	0.57	0.57	0.57

building	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.55	0.57	0.50	0.44	0.46	0.45
Rec	0.27	0.34	0.42	0.73	0.67	0.72
F1	0.36	0.42	0.46	0.55	0.54	0.55

vegetation	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.54	0.51	0.45	0.41	0.43	0.41
Rec	0.14	0.35	0.46	0.64	0.61	0.64
F1	0.21	0.42	0.45	0.50	0.50	0.50

road	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.59	0.58	0.50	0.46	0.47	0.46
Rec	0.35	0.34	0.47	0.75	0.71	0.76
F1	0.44	0.42	0.48	0.57	0.56	0.57

hand	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.33	0.46	0.42	0.42	0.42	0.42
Rec	0.06	0.31	0.40	0.69	0.68	0.69
F1	0.10	0.37	0.41	0.53	0.52	0.52

urban	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.53	0.51	0.47	0.46	0.47	0.46
Rec	0.25	0.41	0.45	0.70	0.67	0.70
F1	0.34	0.46	0.46	0.56	0.55	0.55

crowd	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.78	0.54	0.50	0.42	0.43	0.42
Rec	0.03	0.32	0.49	0.71	0.68	0.71
F1	0.06	0.40	0.50	0.53	0.53	0.53

person	**SVM**	**DT**	**NN**	**SVM**	**DT**	**NN**
Pre	0.61	0.54	0.48	0.44	0.44	0.44
Rec	0.32	0.39	0.45	0.69	0.68	0.69
F1	0.42	0.45	0.47	0.53	0.53	0.53

IV. CONCLUSION

In this paper, the MCA-based data pruning framework using the transaction weights based on correlation is proposed to handle multimedia semantic retrieval related problems such as the large data size and the imbalanced data in a multimedia database. We utilize the functionality of MCA to measure the correlation between the extracted low-level audio-visual features and the classes to infer the target high-level concepts, and consider the correlation information as the feature-value pair weight to infer the transaction weights for the data instances. The training data set is pruned by using these transaction weights. Meanwhile, the testing data set could be weighted and pruned as well so that the computational cost is reduced not only when building the model but also when applying the classifiers. The algorithm is able to automatically and adaptively determine the thresholds for the weights so that only the best representative feature-value pairs would be weighted, and the thresholds for the total transaction weights to determine whether the data instance should be pruned or not. The TRECVID 2007 and 2008 benchmark data (18 concepts) is used to evaluate the classification performance of three well-known and commonly used classifiers trained by the original and pruned training data sets. The experimental results show that our proposed framework performs well in improving the accuracy of the detection of the high-level concepts, in reducing the complexity of the data sets, and in achieving robustness tests for both balanced and imbalanced data sets.

ACKNOWLEDGMENT

For Shu-Ching Chen, this research was carried out in part by a grant from the National Oceanic and Atmospheric Administration (NOAA).

REFERENCES

[1] R. Datta, D. Joshi, J. Li, and J. Z. Wang, “Image retrieval: Ideas, influences, and trends of the new age,” *ACM Computing Surveys (CSUR)*, vol. 40, no. 2, pp. 1–60, 2008.

[2] C. G. M. Snoek and M. Worring, “Concept-based video retrieval,” *Foundations and Trends in Information Retrieval*, vol. 2, no. 4, pp. 215–322, 2008.

[3] M.-L. Shyu, S.-C. Chen, Q. Sun, and H. Yu, “Overview and future trends of multimedia research of content access and distribution,” *International Journal of Semantic Computing*, vol. 1, no. 1, pp. 29–66, March 2007.

[4] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, “Content-based multimedia information retrieval: State of art and challenges,” *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 2, no. 1, pp. 1–19, 2006.

[5] N. Japkowicz and S. Stephen, “The class imbalance problem: A systematic study,” *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[6] A. Estabrooks, T. Jo, and N. Japkowicz, “A multiple resampling method for learning from imbalanced data sets,” *Computational Intelligence*, vol. 20, no. 1, pp. 18–36, February 2004.

[7] G. M. Weiss and F. Provost, “Learning when training data are costly: The effect of class distribution on tree induction,” *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, October 2003.

[8] A. Kohrs and B. Merialdo, “Improving collaborative filtering with multimedia indexing techniques to create user-adapting web sites,” in *ACM International Conference on Multimedia*, 1999, pp. 27–36.

[9] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[10] L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen, “Video semantic concept discovery using multimodal-based association classification,” in *IEEE International Conference on Multimedia and Expo (ICME07)*, July 2007, pp. 859–862.

[11] A. Angelova, Y. Abu-Mostafa, and P. Perona, “Pruning training sets for learning of object categories,” in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR05)*, June 2005, pp. 494–501.

[12] A. Vezhnevets and O. Barinova, “Avoiding boosting overfitting by removing confusing samples,” in *European Conference on Machine Learning (ECML07)*, 2007, pp. 430–441.

[13] G. Herman, G. Ye, J. Xu, and B. Zhang, “Improving object detection by removing noisy samples from training sets,” in *ACM International Conference on Multimedia Information Retrieval (MIR08)*, 2008, pp. 329–335.

[14] S. Wang, M. Dash, L.-T. Chia, and M. Xu, “Efficient sampling of training set in large and noisy multimedia data,” *ACM Transactions on Multimedia Computing, Communications and Applications (TOMCCAP07)*, vol. 3, no. 3, pp. 1–26, 2007.

[15] J. Chen, R. Wang, S. Yan, S. Shan, X. Chen, and W. Gao, “Enhancing human face detection by resampling examples through manifolds,” *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 137, pp. 1017–1028, November 2007.

[16] Y. Tao, X. Xiao, and S. Zhou, “Mining distance-based outliers from large databases in any metric space,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD06)*, 2006, pp. 394–403.

[17] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: Identifying density-based local outliers,” *SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000.

[18] H. Xiong, P. Gaurav, M. Steinbach, and K. Vipin, “Enhancing data analysis with noise removal,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 304–319, March 2006.

[19] S. Sarawagi, V. S. Deshpande, and S. Kasliwal, “Efficient top-k count queries over imprecise duplicates,” in *International Conference on Extending Database Technology (EDBT09)*, 2009, pp. 450–461.

[20] L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen, “Correlation-based video semantic concept detection using multiple correspondence analysis,” in *IEEE International Symposium on Multimedia (ISM08)*, December 2008, pp. 316–321.

[21] L. Lin, M.-L. Shyu, G. Ravitz, and S.-C. Chen, “Video semantic concept detection via associative classification,” in *IEEE International Conference on Multimedia and Expo (ICME09)*, July 2009, pp. 418–421.

[22] N. J. Salkind, Ed., *Encyclopedia of Measurement and Statistics*. SAGE Publications, Inc, 2007.

[23] A. F. Smeaton, P. Over, and W. Kraaij, “Evaluation campaigns and TRECVid,” in *ACM International Workshop on Multimedia Information Retrieval (MIR06)*, October 2006, pp. 321–330.

[24] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. Morgan Kaufmann, June 2005.